SLS-CS_12-09

High Data Rate (Gbps) Coding Architecture

V. Sank, H. Garon - NASA/GSFC/MEI

W. Fong, W. Horne – NASA/GSFC

High Data Rate (Gbps) Coding Architecture

Purpose

- to avoid having a transmitter encoder built in a way that is incompatible with the ground receiver decoder.

Introduction

- Inherent with any system is the problem of component rate limits. In addition to the encoder and decoder, the circuits that compose the I and Q or higher order modulation channels, hit a rate limit. If there was only one way to architect the system, there would be little concern. The problem is that there are several ways to build the system and the transmitter must have the same structure as the receiver.
- As we push to very high rate Gbps links, we believe that CCSDS should set standards on how encoders and decoders should be multiplexed. Vendors are ahead of CCSDS and several have already chosen an architecture.
- Our intent here, at a minimum, is to make the community aware that a *laissez-faire* approach towards standards is a poor choice. Since there are multiple structures possible, every user must clearly specify their intended structure and must insure that the encoder and decoder are compatible.

High Data Rate (Gbps) Coding Architecture

Background

- In days past, in a case with a convolutional code, and rate limited decoders,

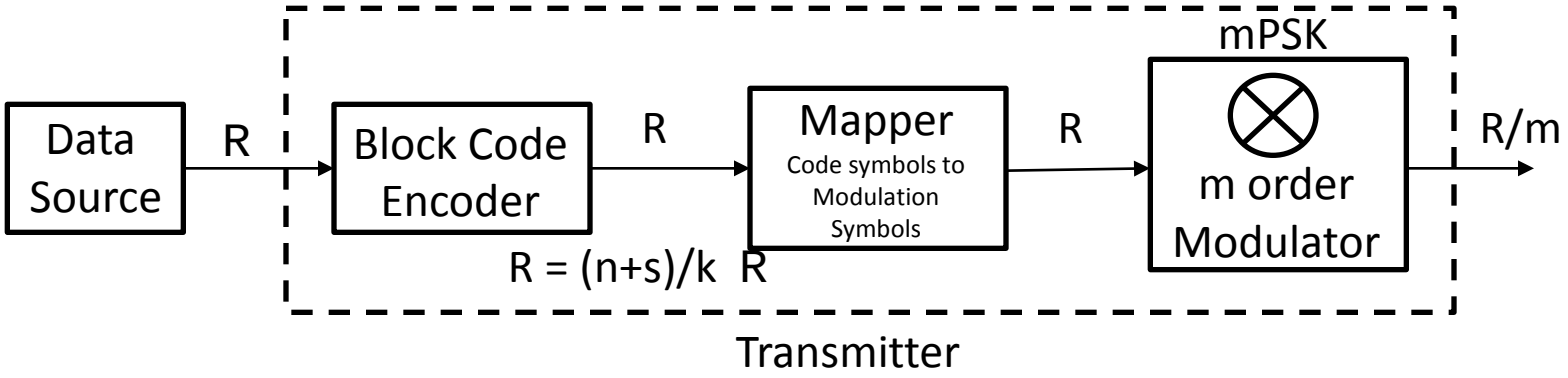
Several decoder were required. Not realizing that there was more than one way to architect the system, the vendor chose what was thought to be the only or obvious structure, and built spacecraft. It was not until after CDR and after the spacecraft was built, that we asked the vendor how he knew what the ground station design was. It turned out that the spacecraft was built wrong and the transmitter encoder needed to be disassembled and redesigned.

Scope

- This presentation is not intended to be a design document or specification.
- The following diagrams do not consider the latencies, memory buffers and frame synchronization that will be required in a formal specification or real system.

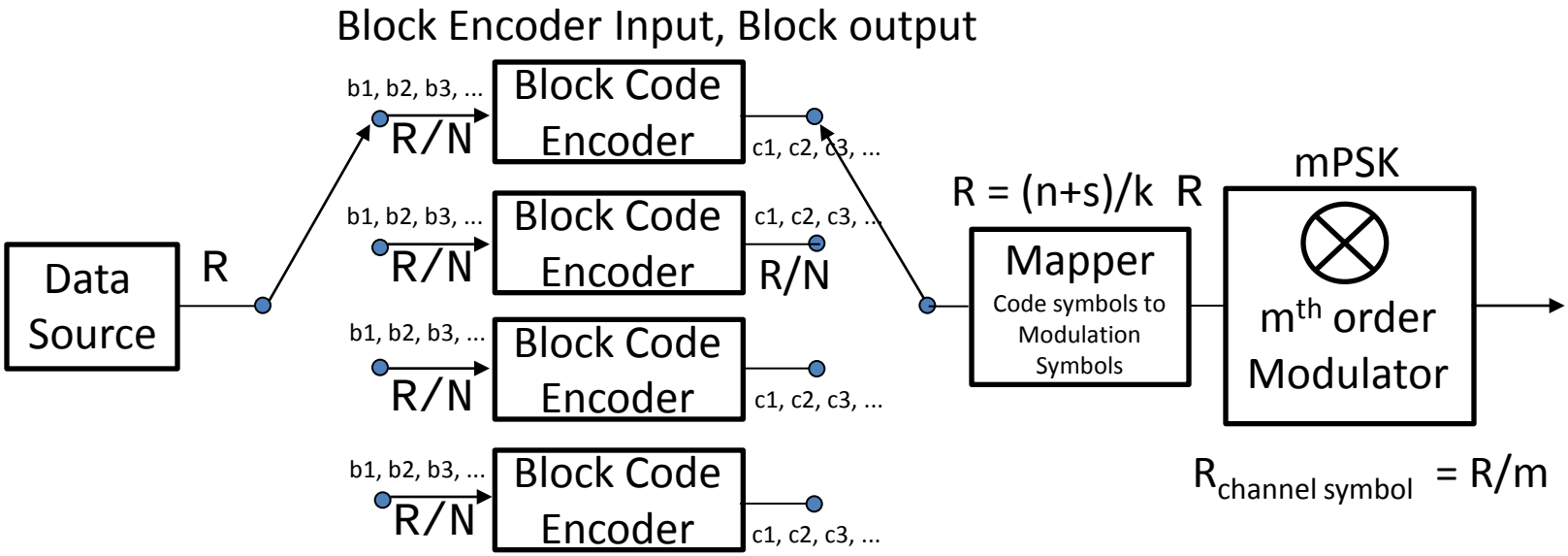
High Data Rate (Gbps) Coding Architecture

R = data rate (from C&DH)
R = coded data rate (code symbol rate) = R_s
N = number of encoders
k = codeword message size
n = codeword size including parity
m = modulation order (ie. m=3 => 8PSK)
s = size of ASM in same units as k and n (bits, nibbles, bytes)



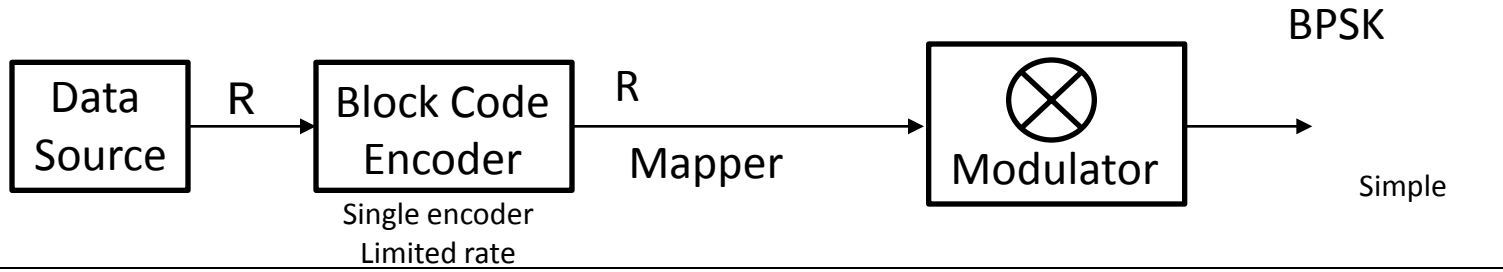
- Diagrams are shown from the point of view of the transmitter.
- The structure of the decoder must complement the transmitter, except as noted.

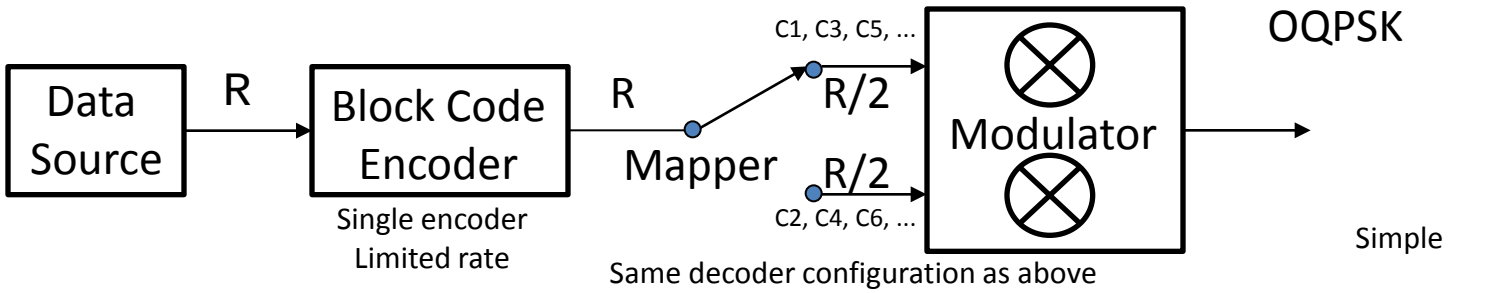
Recommended Architecture



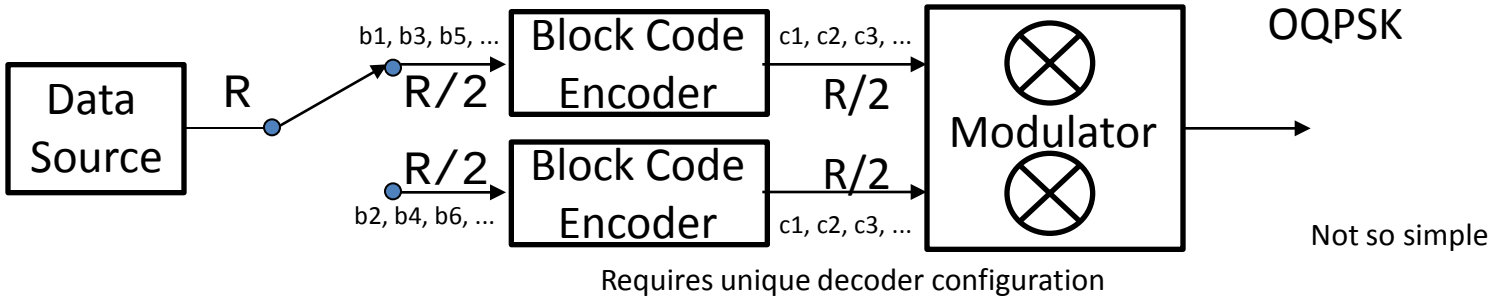
- An encoder is loaded with the full message size, k, before data starts flowing into the next encoder. (We believe receivers from two vendors are configured this way.)
- Data on the physical channel appears the same as it would if there was a single encoder that runs at the full rate.
- Encoding is generally easier than decoding and encoders generally can run faster than decoders. This architecture has the advantage that the number of encoders and decoders need not be the same.

Encoder, Simple Cases



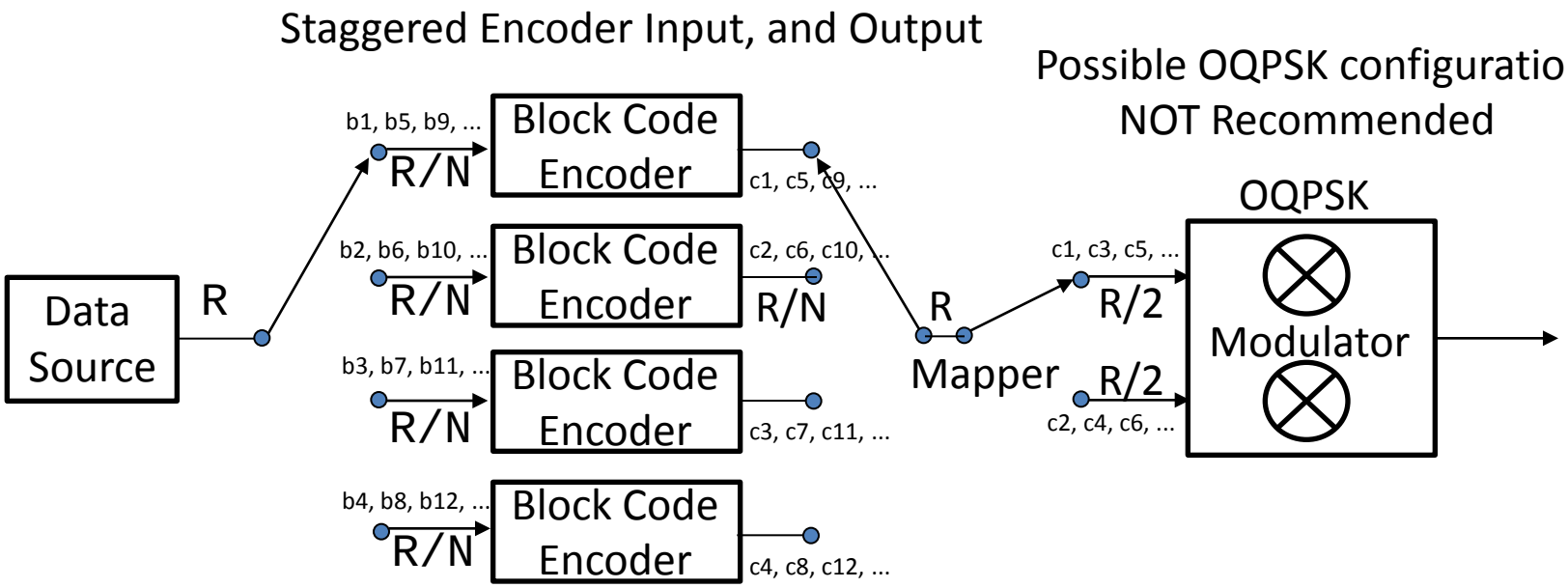


Staggered Encoder Input Independent Encoders on the I and Q channel



When more than 1 encoder, input is shown as either b1 b2 b3 which means that a constant stream of bits are input, or b1 b3 b5 and b2 b4 b6 which means that the bits are staggered from one encoder to the other.

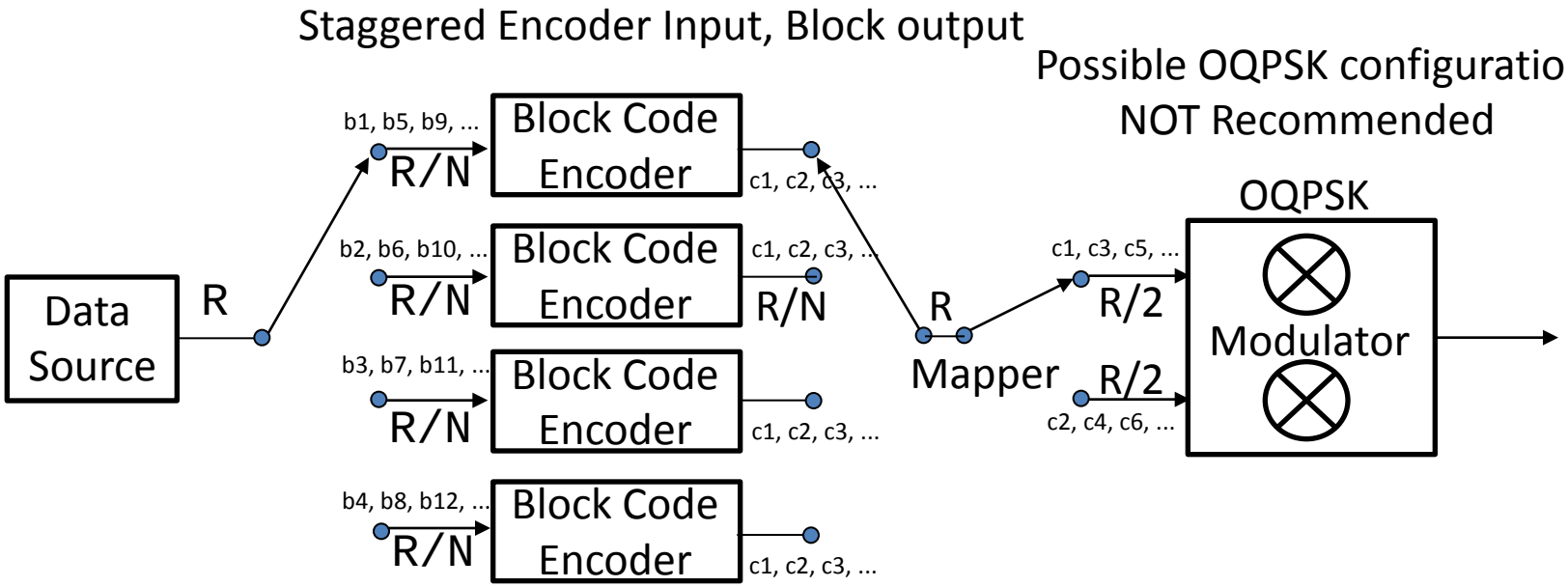
Quad Encoder, OQPSK version of recommended configuration



Not the recommended configuration but it has an advantage.

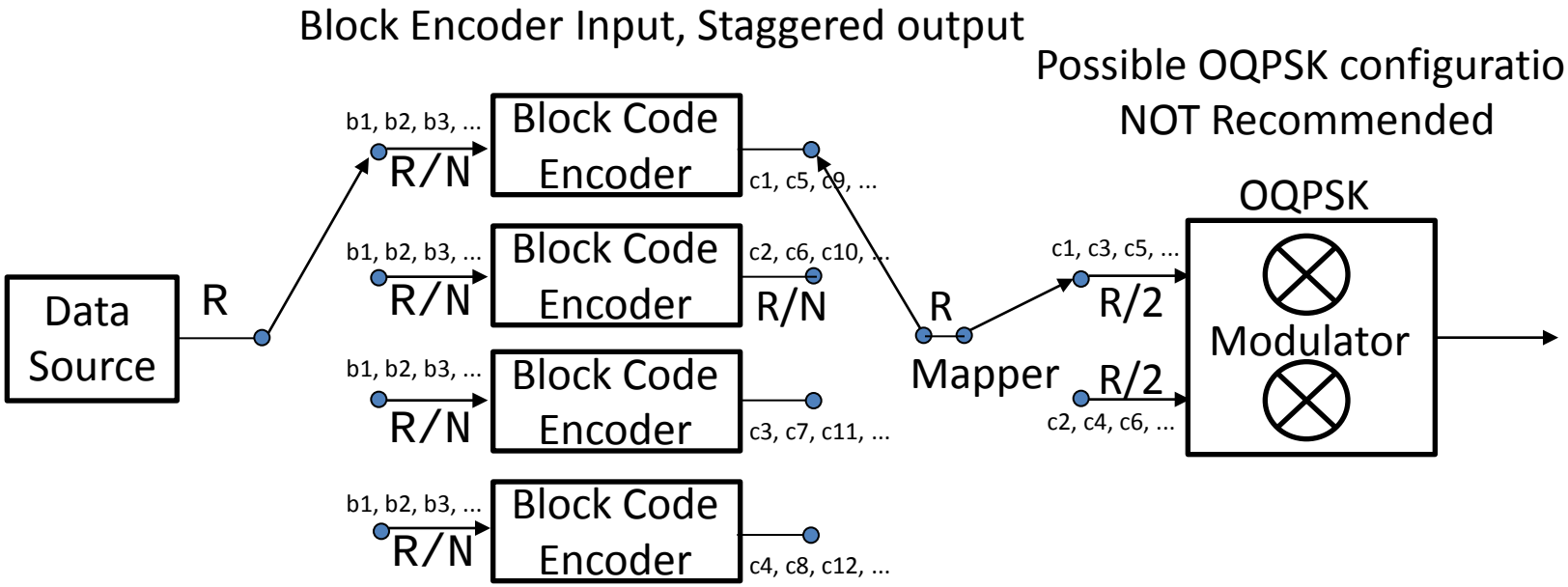
- M ≡ average number of codeword errors that can be corrected over all codewords.
- On average, this configuration will correct a burst of N*M bits because consecutive bits on the physical channel are from different encoders.
- The Reed Solomon Interleave has the same configuration.
- An implementation that is expecting a high burst noise background may prefer such a custom design with staggered encoder output.

Quad Encoder, OQPSK possible configuration



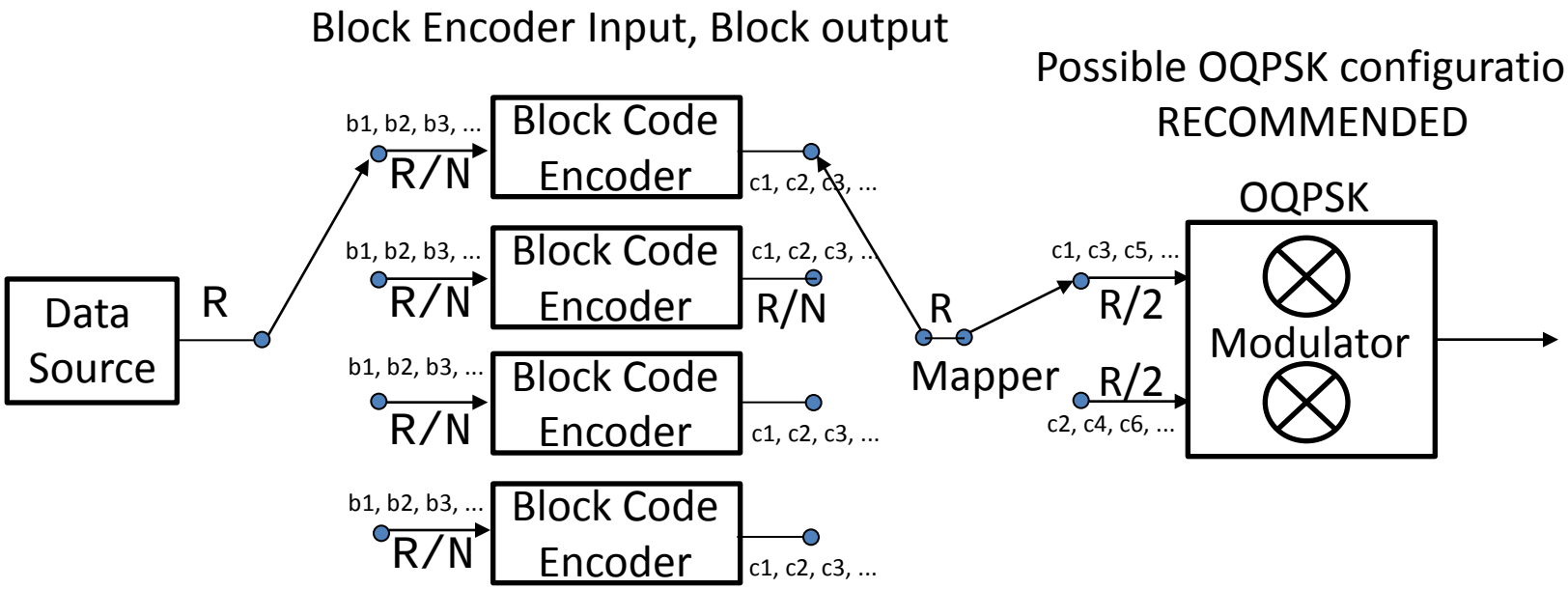
- This architecture has the disadvantage that an errored codeword would spread errored bits across several data transfer frames.
- On average, this configuration will correct a burst of N bits, not N channel symbols.

Quad Encoder, OQPSK possible configuration



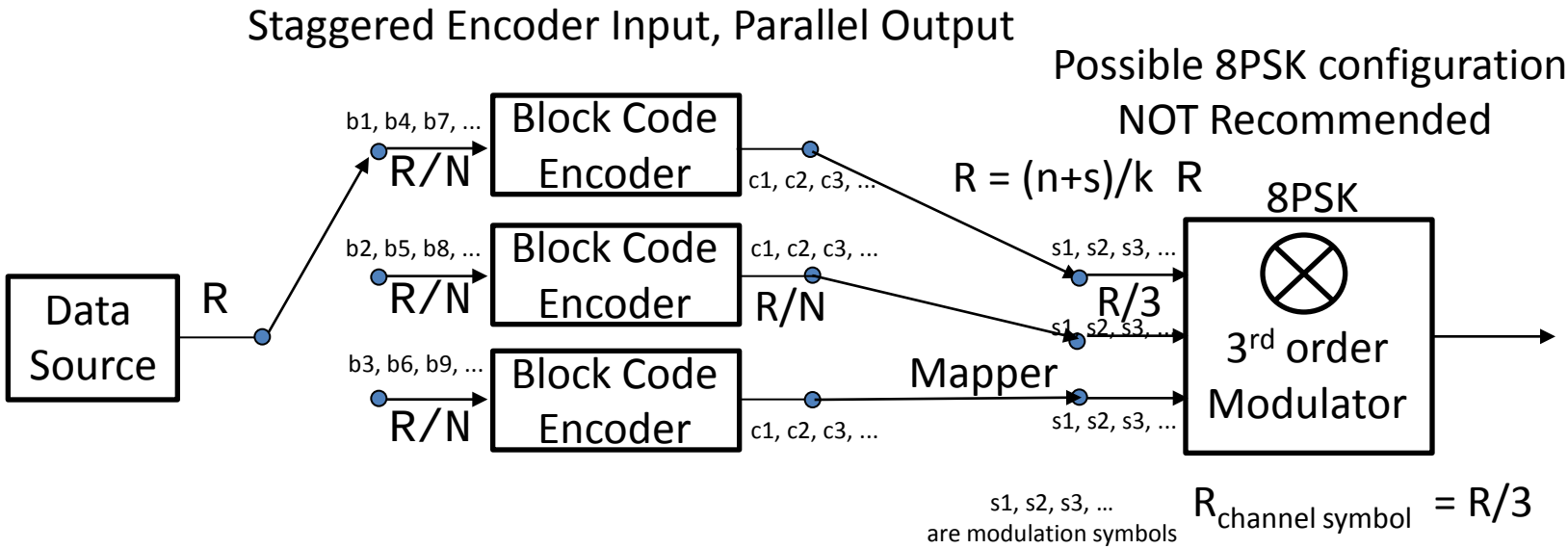
- This architecture has the DISadvantage that the number of encoders and decoders must be the same
- The decoder output must select the which decoder to consider the first.
- On average, this configuration will correct a burst of N*M bits because consecutive bits on the physical channel are from different encoders.

Quad Encoder, OQPSK possible configuration



- Physical channel appears the same as single encoder that runs at full rate.
- Encoding is generally easier than decoding and encoders generally can run faster than decoders
- This architecture has the advantage that the number of encoders and decoders need not be the same.

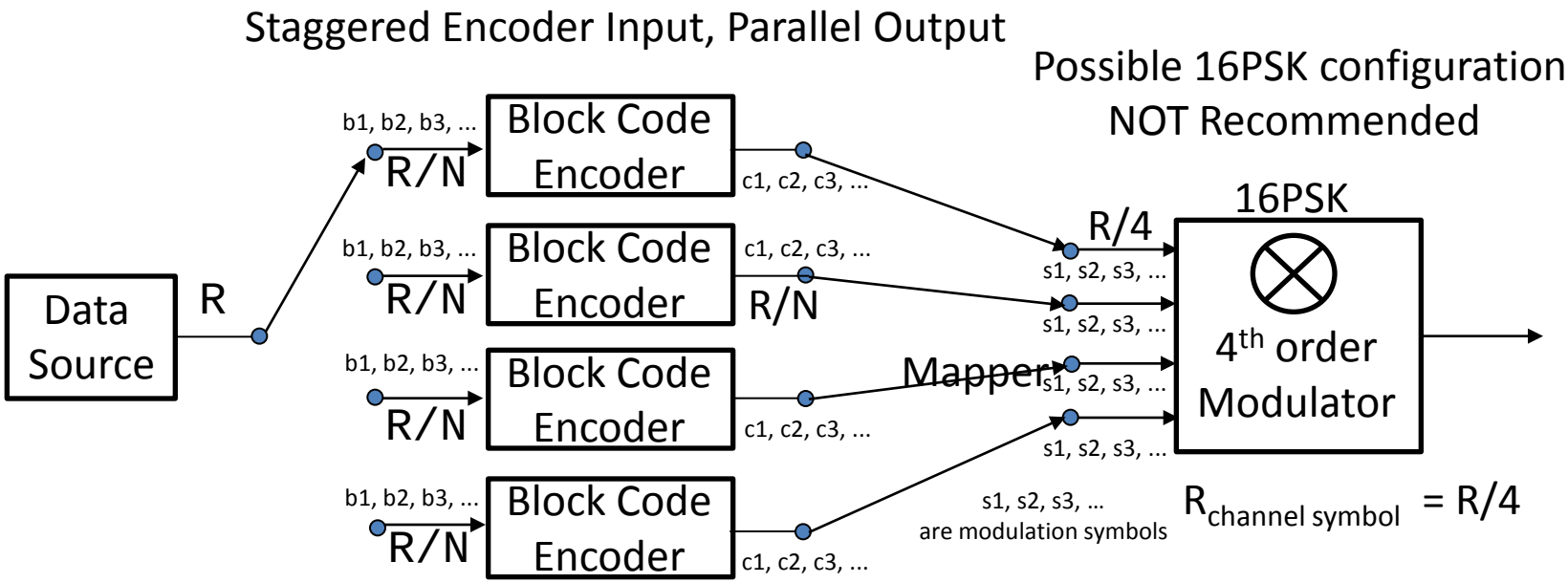
Triple Encoder, 8PSK possible configuration



- This architecture has the advantage that the number of encoders and decoders need not be the same. But the number on each channel must be the same.
- A cover code can be used to synchronize the decoders and resolve ambiguity.
- On average, this configuration will correct a burst of N channel symbols, N*M bits, because consecutive bits on the physical channel are from different encoders.

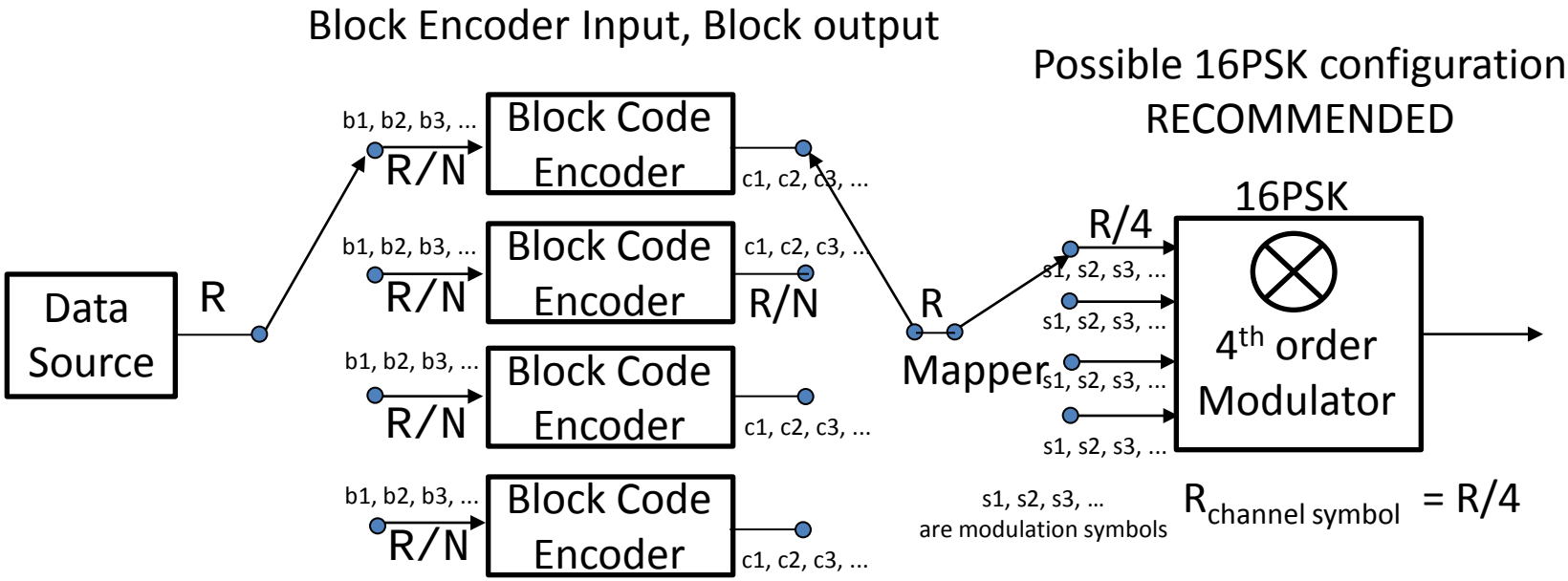
Cover Code Example: Invert input to one of the encoders.

Quad Encoder, 16PSK possible configuration



- This architecture has the advantage that the number of encoders and decoders need not be the same. But the number on each channel must be the same.
- A cover code can be used to synchronize the decoders and resolve ambiguity.
- On average, this configuration will correct a burst of N channel symbols, N*M bits, because consecutive bits on the physical channel are from different encoders.

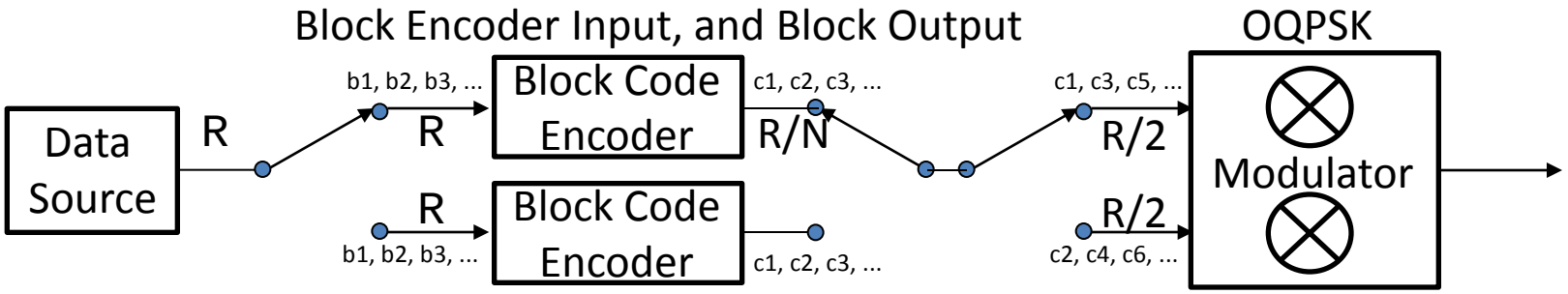
Quad Encoder, 16PSK possible configuration



- This architecture has the advantage that the number of encoders and decoders need not be the same.
- On average, this configuration will correct a burst of N bits, not N channel symbols
- A cover code can be used to synchronize the decoders and resolve ambiguity.

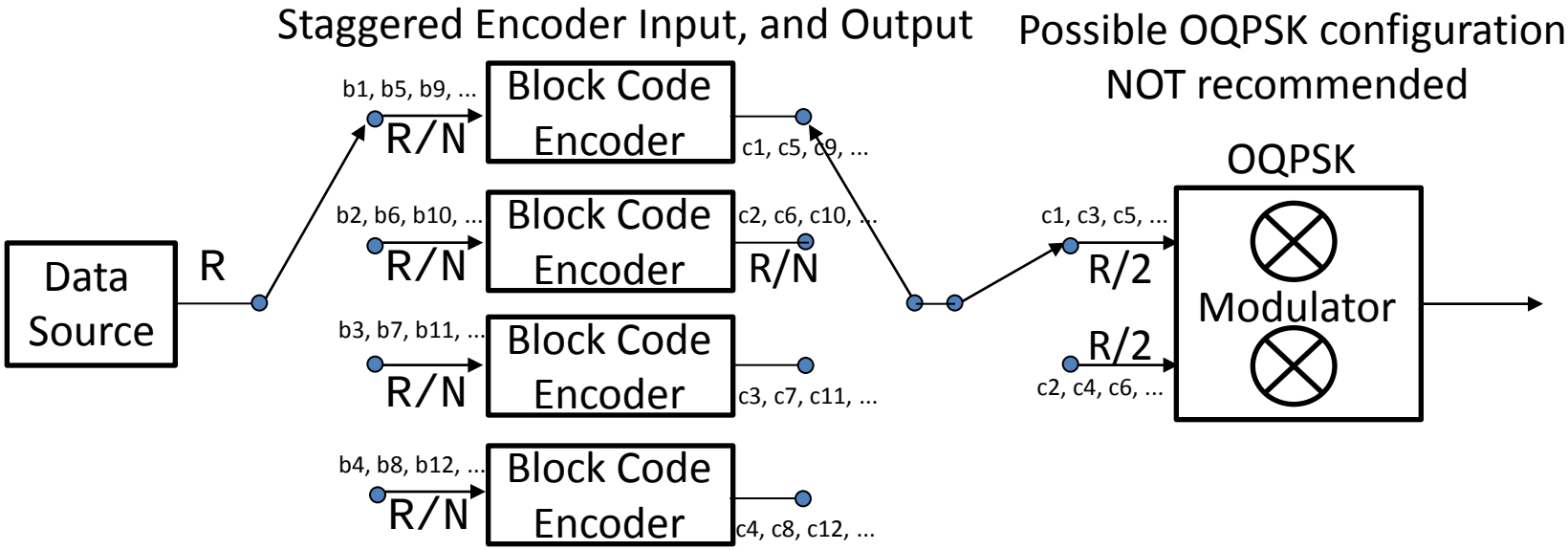
Back Up

Dual Encoder, OQPSK version of Recommended configuration



Requires start up latency, but Effective rate is R/N

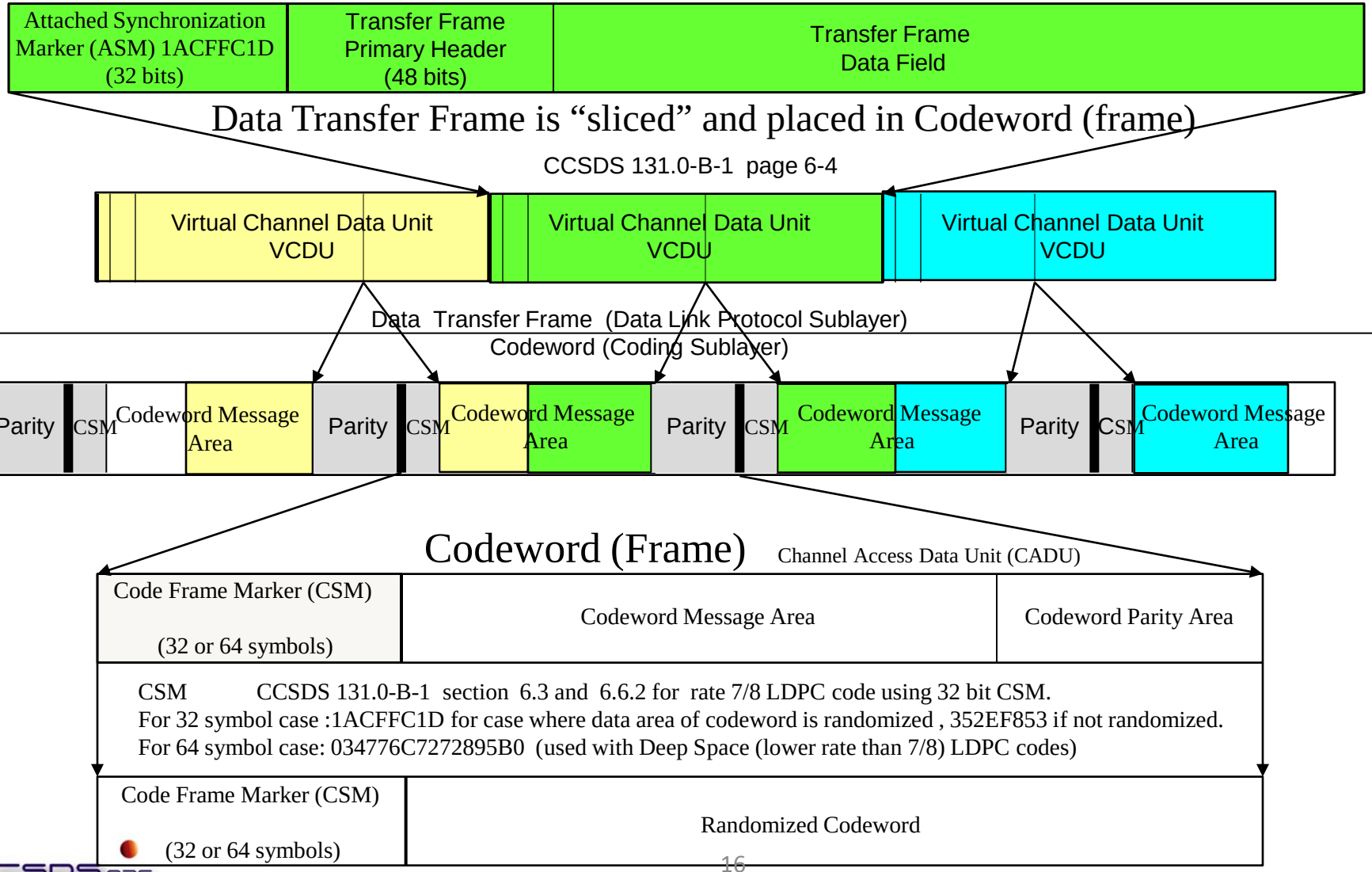
Physical channel appears the same as single encoder that runs at full rate.



Low start up latency, but this should not be an issue at Gbps rates

Sliced (Asynchronous) Data Transfer Frame and Code Frame

CCSDS 131.0-B-1, 6.3 CCSDS 732.0-B-2 page 4-2 **Asynchronous** here means that the Transfer frame and Code frame are different length.



ASM and CSM

Using the same pattern for both the Data Transfer Frame ASM and codeword ASM (CSM):
LDPC is a "systematic" code, which means that the message data Transfer frame and its ASM is not changed when encoded, only parity is added. Randomization is applied after the coding is done so the data frame ASM gets randomized.
Are there ever instances when we want to turn off the CCSDS randomizer?
The basic assumption of CCSDS randomization is that it is either always ON or always OFF. But during I&T or during debug it may be off.
Even if it is turned off, this doesn't preclude using the same pattern for both the ASM and CSM.
The distance between each ASM will be different for ASM and CSM. This allows the frame synchronizer to find the correct frame. If the two frames were the same length, they could and should be made synchronous. So we assume that they are not the same length.

Decoding

When using a Cortex XXL ground receiver there are several options on how the data is handed from the code sub level to the data sub level. All code frames (codewords) can be handed to the next higher level, the data level, even if the decoder fails to correct all of the errors. Or, the receiver can be set up to drop the code frames that are not fully decoded.
When the code frame and data frame are not the same length and aligned (asynchronous), a failure to correct all errors at the decoding level of a single code frame, can result in one or many data frames with errors, depending on the size of the code and data frame.
If all code frames are passed to the data level, it is suggested that the CCSDS CRC is used so that at the data sublevel, the corrupted data frames can be identified. For high data rate missions, greater than about 150 Mbps, it may be difficult to include the CRC since its value needs to be calculated on the fly, as data frames are built and the CCSDS sequence count is determined.
If the CCSDS CRC is not used, it is recommended that the receiver be set to not pass all code frames to the data level. This way the missing bits will cause the data frame synch to drop lock. The operations center may be able to have the missing data frames and a few on both sides retransmitted. It is important to retransmit at least one frame prior to the point where frame synch was lost because the frame just prior to the loss of frame synch may also contain errors.